CLAIMS

WHAT IS CLAIMED IS:

1. A method comprising:

determining whether a thread encountered a scoped breakpoint; and

if the determining is true, halting execution of the thread if the thread previously encountered an entry breakpoint.

2. The method of claim 1, further comprising:

if the determining is true, allowing execution of the thread to continue if the thread did not previously encounter the entry breakpoint.

3. The method of claim 1, wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.

4. The method of claim 1, wherein the entry breakpoint is executed conditionally.

5. An apparatus comprising:

means for determining whether a thread encountered a scoped breakpoint;

means for halting execution of the thread if the thread previously encountered an entry breakpoint and if the determining is true; and

means for allowing execution of the thread to continue if the thread did not previously encounter the entry breakpoint and if the determining is true.

6. The apparatus of claim 5, wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.

7. The apparatus of claim 5, wherein the entry breakpoint is executed conditionally.

8. The apparatus of claim 6, further comprising:

means for allowing execution of the thread to continue upon the thread encountering the entry breakpoint and the end breakpoint.

9. A signal-bearing medium encoded with instructions, wherein the instructions when executed comprise:

saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint;

if a thread encounters the entry breakpoint, saving an identifier of the thread;

if the thread encounters a scoped breakpoint within the region, determining whether the identifier was saved;

if the identifier was saved and the scoped breakpoint was encountered, halting execution of the thread; and

if the identifier was not saved and the scoped breakpoint was encountered, allowing execution of the thread to continue.

10. The signal-bearing medium of claim 9, further comprising:

allowing execution of the thread to continue upon the thread encountering the entry breakpoint; and

allowing execution of the thread to continue upon the thread encountering the end breakpoint.

11. The signal-bearing medium of claim 9, further comprising:

saving a definition of the scoped breakpoint within the region;

12. The signal-bearing medium of claim 9, further comprising:

after the thread encounters the end breakpoint, removing the saved identifier of the thread.

13. A computer system comprising:

a processor; and

memory encoded with instructions, wherein the instructions when executed on the processor comprise:

saving a definition of a region in a program bounded by an entry breakpoint and an end breakpoint,

saving a definition of a scoped breakpoint within the region,

if a thread encounters the entry breakpoint, saving an identifier of the thread,

if the thread encounters the scoped breakpoint within the region, determining whether the identifier was saved,

if the identifier was saved and the scoped breakpoint was encountered, halting execution of the thread, and

if the identifier was not saved and the scoped breakpoint was encountered, allowing execution of the thread to continue.

14. The computer system of claim 13, wherein the instructions further comprise:

allowing execution of the thread to continue upon the thread encountering the entry breakpoint; and

allowing execution of the thread to continue upon the thread encountering the end breakpoint.

15. The computer system of claim 13, wherein the instructions further comprise:

after the thread encounters the end breakpoint, removing the saved identifier of the thread.

16. The computer system of claim 13, wherein the entry breakpoint is part of a conditional construct in the program.

17. A method of configuring a computer, wherein the method comprises:

configuring the computer to determine whether a thread encountered a scoped breakpoint; and

configuring the computer to halt execution of the thread if the thread previously encountered an entry breakpoint and if the thread encountered the scoped breakpoint.

18. The method of claim 17, further comprising:

configuring the computer to allow execution of the thread to continue if the thread did not previously encounter the entry breakpoint and if the thread encountered the scoped breakpoint.

19. The method of claim 17, wherein the scoped breakpoint is within a region bounded by the entry breakpoint and an end breakpoint.

20. The method of claim 17, wherein the entry breakpoint is executed conditionally.